1. Show the memory layout of the following local variables allocated on the runtime Stack taking into consideration the SPARC data type memory alignment restrictions discussed in class. Fill bytes in memory with the appropriate local variable name. For example, if local variable name p takes 4 bytes, you will have 4 p's in the appropriate memory locations. If the local variable is an array, use the name followed by the index number. For example, some number of p0's, p1's, p2's, etc. Place an X in any bytes of padding.

```plaintext
char a;
short b;
int c;
short d[3];
double e;
```

Write the SPARC assembly instructions that would be generated by the following instruction using the above memory layout of the local variables. Make no assumptions about values already loaded into any registers. All local variable accesses must access the local variable space allocated on the Stack. Just give a straightforward suboptimal code generation to perform this operation. **Hint:** This should take 4 instructions.

c = 420420 + c;

________________________________
________________________________
________________________________
________________________________
2. Given the following code for foo(), write an equivalent more highly optimized version in SPARC assembly. Assume:

- a is mapped to local register %l0
- b is mapped to local register %l1
- c is mapped to local register %l2
- x is a global variable allocated in the Data segment and NOT mapped to a register

---

### Oberon

```oberon
VAR x : INTEGER;
PROCEDURE foo( i : INTEGER );
  VAR a, b, c : INTEGER;
BEGIN
  a := 15;
  b := a + 10;
  c := i + b;
  x := c;
END foo;
BEGIN
  foo( 5 );
END.
```

---

### SPARC Assembly

```assembly
.global main, foo
.var x : INTEGER
.section ".data"
align 4
x: .word 0
.section ".text"
foo:
  save %sp, -96, %sp
  mov 15, %l0
  add %l0, 10, %l1
  add %i0, %l1, %l2
  x := c;
  a := x;
  b := x;
  x := x;
  (* Other code may access a,b,c,x *)
END foo;
BEGIN
  foo( 5 );
END.
```

Rewrite only code that is in the bounds of the rectangle.

---

What question would you most like to see on the Final?